

We claim:

- 1 1. A redundancy architecture for network processing systems in a
2 network, the network containing network traffic consisting of a plurality of data
3 packets, the plurality of data packets forming a plurality of flows, the redundancy
4 architecture comprising:
5 at least a first network processing system operable to receive data packets
6 from the network and further operable to send processed data packets back onto the
7 network and including, a processing engine operable to learn and maintain state for
8 particular flows, wherein the processing engine assigns an identifier to one or more
9 of the particular flows and associates each data packet belonging to that flow with the
10 identifier, the processing engine further operable to identify characteristics of one or
11 more of the particular flows and to store those characteristics in a state database in the
12 processing engine, the network processing system further including a network link;
13 and
14 a second network processing system which includes a second processing
15 engine with a second database and a second network link connected to the network
16 link and operable to allow communication between the first network processing
17 system and the second network processing system;
18 wherein the second network processing system is connected in parallel with
19 the first network processing system and is operable to receive all the state entries from
20 the first network processing system, such that when a failure occurs in the first
21 network processing system the network traffic on the first network processing system
22 is switched to the second network processing system.
- 1 2. The redundancy architecture of Claim 1 wherein the first and second
2 network processing systems share the network traffic while maintaining the state
3 entries for both the first and second network processing systems.
- 1 3. The redundancy architecture of Claim 1 wherein the first network
2 processing system processes all the network traffic, while the second network
3 processing system maintains all the state entries for the first network processing
4 system.

1 4. The redundancy architecture of Claim 1 further comprising at least a
2 third network processing system connected in parallel with the first and second
3 network processing systems, wherein the second network processing system
4 maintains the state entries for both the first and third network processing systems,
5 such that when a failure occurs in either the first or third network processing system
6 the network traffic on the failed network processing system is switched to the second
7 network processing system.

1 5. The redundancy architecture of Claim 1 wherein first and second
2 network processing systems are co-located.

1 6. The redundancy architecture of Claim 5 wherein the network
2 processing system are configured in a dual port configuration to provide protection
3 against link failures.

1 7. The redundancy architecture of Claim 1 wherein the first and second
2 network processing systems are remotely located.

1 8. The redundancy architecture of Claim 1 wherein the network link is an
2 Ethernet connection.

1 9. The redundancy architecture of Claim 1 wherein the first and second
2 network processing systems are automatically returned to a redundant state when the
3 first network processing system returns to an operational state.

1 10. A redundancy architecture for network processing systems in a
2 network, the network containing network traffic consisting of a plurality of data
3 packets, the plurality of data packets forming a plurality of flows, the redundancy
4 architecture comprising:
5 a first network processing system operable to receive a first set of data packets
6 from the network and further operable to send the first set of processed data packets
7 back onto the network and including, a processing engine operable to learn and
8 maintain state for a first set of flows corresponding to the first set of data packets,
9 wherein the processing engine assigns an identifier to one or more of the first set of
10 particular flows and associates each of the first set of data packets belonging to that
11 flow with the identifier, the processing engine further operable to identify
12 characteristics of one or more of the first set of particular flows and to store those
13 characteristics in a first state database in the processing engine, the network
14 processing system further including a network link; and
15 a second network processing system operable to receive a second set of data
16 packets from the network and further operable to send the second set of processed
17 data packets back onto the network and including, a processing engine operable to
18 learn and maintain state for a second set of flows corresponding to the second set of
19 data packets, wherein the processing engine assigns an identifier to one or more of
20 the second set of particular flows and associates each of the second set of data packets
21 belonging to that flow with the identifier, the processing engine further operable to
22 identify characteristics of one or more of the second set of particular flows and to
23 store those characteristics in a second state database in the processing engine, the
24 network processing system further including a network link;
25 wherein the first and second network processing systems are connected in
26 parallel such that substantially all the first state database is created in the second
27 network processing system and all of the second state database is created in the first
28 network processing system, such that when a failure occurs in the first network
29 processing system the first set of data packets is switched to the second network
30 processing system.

1 11. The redundancy architecture of Claim 10 wherein the first and second
2 network processing systems share the network traffic while maintaining the state
3 entries for both the first and second network processing systems.

1 12. The redundancy architecture of Claim 10 wherein the first network
2 processing system processes all the network traffic, while the second network
3 processing system maintains all the state entries for the first network processing
4 system.

1 13. The redundancy architecture of Claim 10 further comprising at least a
2 third network processing system connected in parallel with the first and second
3 network processing systems, wherein the second network processing system
4 maintains the state entries for both the first and third network processing systems,
5 such that when a failure occurs in either the first or third network processing system
6 the network traffic on the failed network processing system is switched to the second
7 network processing system.

1 14. The redundancy architecture of Claim 10 wherein first and second
2 network processing systems are co-located.

1 15. The redundancy architecture of Claim 14 wherein the network
2 processing system are configured in a dual port configuration to provide protection
3 against link failures.

1 16. The redundancy architecture of Claim 10 wherein the first and second
2 network processing systems are remotely located.

1 17. The redundancy architecture of Claim 10 wherein the network link is
2 an Ethernet connection.

18. The redundancy architecture of Claim 10 wherein the first and second network processing systems are automatically returned to a redundant state when the first network processing system returns to an operational state.